Reinforcement Learning: An Introduction

Summary Notes

Mrinank Sharma

March 26, 2020

1 Introduction

Elements of Reinforcement Learning

Other than the **agent** and the **environment**, there are four main subelements of an RL system:

1. The *policy* maps perceived states of the environment to actions. This could be a lookup table or a search process.
2. The *reward signal* defines the goal of a reinforement learning problem. The environment sends rewards to an agent that has the sole objective of maximising this reward.
3. The *value function* specifies what is good in the long run. The *value* of a state is the total amount of reward an agent can expect to accumulate over teh future starting from that state.
4. Some RL systems have *models* of the environment that can mimic the behaviour of the environment. Models are used for planning.

Rewards and Value Functions

Whilst rewards determine the immediate, intrisic desirability of environmental states, values indicate the long-term desirability of states after taking into account which states are likely to follow. We choose actions based on values, even though values are derived from rewards.

Evolutionary Methods

It is possible to learn policies without estimating value functions by using evolutionary methods. However, they tend to ignore much of the useful structure and are not well suited for RL tasks as they ignore relevant information, such as the states which are passed and the actions which are selected.

2 Multi-armed Bandits

> *The most important feature distinguishing reinforcement learning from other types of learning is that it uses training information that **evaluates** the actions taken rather than **instructs** by giving correct actions.*

This form of feedback creates the need for active exploration.

k-armed bandits

Definition 2.1 (k-armed Bandit) *In the k-armed bandit problem, you are repeatedly faced with a choice among k different actions. After each choice, a numerical reward is received from a **stationary** probability distribution associated with the selected action. The objective is to maximise the expected total reward over some time period.*

Value

The value of action a is the expected reward given a was chosen:

$$q_*(a) = \mathbb{E}[R_t | A_t = a]. \tag{1}$$

If we knew the values, problem solved - just pick the action with the maximum value. However, we do not know these. Denote the *estimated* value as $Q_t(a)$. A simple way to estimate the action-value is a *sample average*, which can be calculated incrementally.

Reward Variance

Note that purely greedy methods are best if there is no noise in the reward signal. With ϵ-greedy algorithms, the best value of ϵ will depend on the amount of noise, which

determines the number of samples for the sample average to converge. However, if the problem is *non-stationary*, we still need to explore.

Non-stationary Problems

If the bandit is non-stationary, we might use the following rule for value estimates:

$$Q_{n+1}(a) = Q_n(a) + \alpha_n(a)\left[R_n - Q_n(a)\right]. \tag{2}$$

Convergence Guarantees

It is known that convergence with probability 1 happens if the following conditions hold:

$$\sum_{n=1}^{\infty} \alpha_n(a) = \infty, \text{ and } \sum_{n=1}^{\infty} \alpha_n(a)^2 < \infty. \tag{3}$$

The first condition ensures the steps are sufficiently large to overcome an initial condition and the second condition ensures that the steps become small enough to give convergence. However, step sizes which meet this conditions are rarely used in applications because they seem to converge slowly or require considerable tuning.

With this techniques, the initialisation effectively becomes an additional parameter which needs to be chosen. They can be used to provide prior knowledge and can encourage exploration, for example, by being optimistic. However, this sort of exploration is not useful for non-stationary problems (the task changes creates a renewed need for exploration).

UCB

The intuition behind UCB is that its better, when exploring, to select among the non-greedy actions which have the *potential* for being optimal. To do this, we maximise an estimate of the arm's value summed with some sort of uncertainty estimate. UCB often performs well but can be difficult to extent to RL from bandits.

We don't have to use value based methods. Instead, we could directly learn a preference over the actions and modify our preferences to give us higher rewards e.g., increasing our preference for an action if selecting it lead to a better outcome than expected.

Contextual Bandits

Definition 2.2 (Contextual Bandits) *In a contextual bandit, the agent is faced with a series of multi-armed bandit problems, each associated with some information i.e., the agent has information about which bandit problem is being faced at any timestep.*

Contextual bandits are intermediate between multi-armed bandits and the full RL problem; they involve learning a policy (i.e., a mapping from state information to actions) but similar to k-armed bandits, **the action selected influences only the immediate reward and not the next situation**.